Uma sessão HTTP típica

Em protocolos cliente-servidor, como o HTTP, sessões consistem em três fases:

1. O cliente estabelece uma conexão TCP (ou a conexão apropriada se a camada de transporte não for TCP).
2. O cliente envia sua requisição e aguarda a resposta.
3. O servidor processa a requisição, envia de volta sua resposta, fornecendo um código de status e os dados apropriados.

A partir do **HTTP/1.1**, a conexão não é mais encerrada após a conclusão da terceira fase, e o cliente agora tem permissão para fazer novas requisições: isso significa que as fases dois e três podem ser realizadas qualquer número de vezes.

Estabelecendo uma conexão

Em protocolos cliente-servidor, é o cliente que estabelece a conexão.

Abrir uma conexão no HTTP significa iniciar uma conexão na camada de transporte subjacente — geralmente o **TCP**.

Com TCP, a porta padrão para um servidor HTTP em um computador é a **porta 80**. Outras portas também podem ser usadas, como **8000** ou **8080**. A URL de uma página a ser buscada contém tanto o nome do domínio quanto o número da porta, embora este último possa ser omitido se for 80. Veja a referência de URL para mais detalhes.

Nota: O modelo cliente-servidor não permite que o servidor envie dados ao cliente sem uma requisição explícita. No entanto, várias APIs Web permitem esse caso de uso, incluindo a **Push API**, **Server-sent events** e a **WebSockets API**.

Enviando uma requisição do cliente

Uma vez estabelecida a conexão, o **user-agent** pode enviar a requisição (um *user-agent* normalmente é um navegador, mas pode ser qualquer outra coisa, como um rastreador).

Uma requisição do cliente consiste em diretivas de texto separadas por **CRLF** (*carriage return* seguido por *line feed*), divididas em três blocos:

1. A primeira linha contém um **método de requisição**, seguido por seus parâmetros:
    - o caminho do documento, como uma URL absoluta sem o protocolo ou nome de domínio;
    - a versão do protocolo HTTP.
2. As linhas seguintes representam os **cabeçalhos HTTP**, que fornecem ao servidor informações sobre o tipo de dado apropriado (por exemplo, idioma, tipos MIME), ou outros dados que alteram seu comportamento (por exemplo, não enviar resposta se já estiver em cache). Esses cabeçalhos HTTP formam um bloco que termina com uma **linha em branco**.
3. O bloco final é um **bloco de dados** opcional, geralmente usado pelo método **POST**.

Exemplo de requisição

Buscando a página raiz de developer.mozilla.org e informando ao servidor que o *user-agent* prefere a página em francês, se possível:

```makefile
HTTP
GET / HTTP/1.1
Host: developer.mozilla.org
Accept-Language: fr
```

Observe a linha em branco final — ela separa o bloco de cabeçalhos do bloco de dados. Como não há um Content-Length fornecido em um cabeçalho HTTP, esse bloco de dados é apresentado vazio, marcando o fim dos cabeçalhos, permitindo que o servidor processe a requisição assim que receber essa linha em branco.

Exemplo com envio de dados de formulário

```perl
HTTP
POST /contact_form.php HTTP/1.1
Host: developer.mozilla.org
Content-Length: 64
Content-Type: application/x-www-form-urlencoded

name=Joe%20User&request=Send%20me%20one%20of%20your%20catalogue
```

Métodos de requisição

O HTTP define um conjunto de **métodos de requisição** que indicam a ação desejada sobre um recurso. Embora também possam ser substantivos, esses métodos são às vezes chamados de *verbos HTTP*. Os mais comuns são **GET** e **POST**:

- O método **GET** solicita uma representação dos dados do recurso especificado. Requisições com GET devem apenas **recuperar dados**.
- O método **POST** envia dados ao servidor para que ele **modifique seu estado**. Este é o método frequentemente usado em **Formulários HTML**.

Estrutura de uma resposta do servidor

Após o *user-agent* ter enviado sua requisição, o servidor web a processa e, finalmente, retorna uma resposta. De forma semelhante à requisição do cliente, a resposta do servidor é composta de diretivas de texto separadas por CRLF, divididas em três blocos:

1. A **primeira linha**, chamada linha de status, consiste em uma confirmação da versão HTTP usada, seguida por um **código de status** de resposta (e seu breve significado em texto legível por humanos).
2. As linhas seguintes representam **cabeçalhos HTTP específicos**, fornecendo ao cliente informações sobre os dados enviados (por exemplo, tipo, tamanho, algoritmo de compressão utilizado, dicas de cache). Assim como os cabeçalhos da requisição do cliente, esses cabeçalhos formam um bloco encerrado com uma **linha em branco**.
3. O **bloco final** é um **bloco de dados**, que contém os dados opcionais.

Exemplos de respostas

Resposta de sucesso de uma página web:

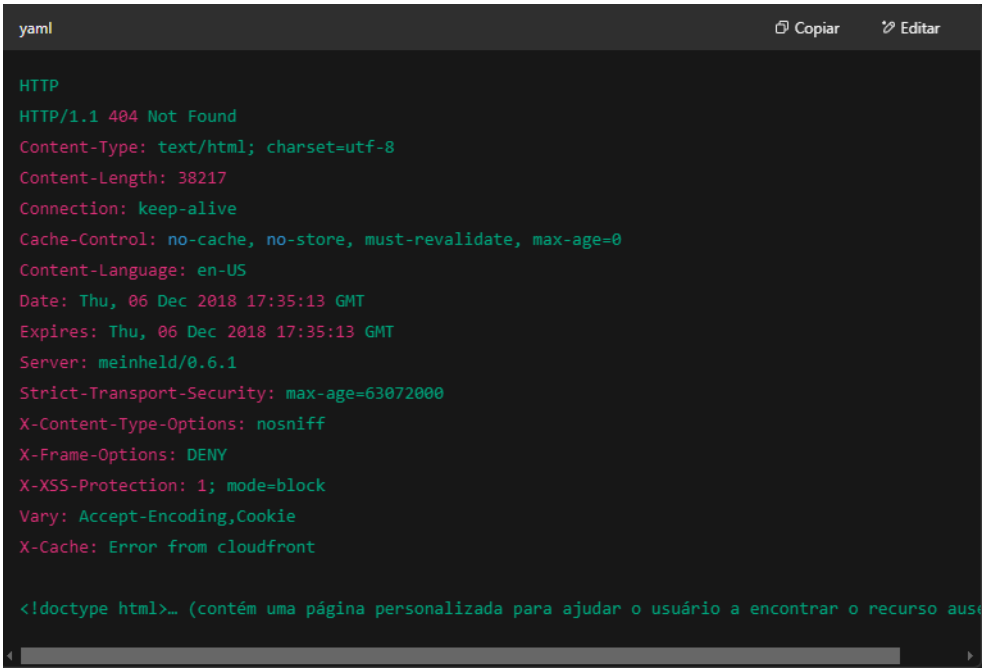
```
HTTP
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 55743
Connection: keep-alive
Cache-Control: s-maxage=300, public, max-age=0
Content-Language: en-US
Date: Thu, 06 Dec 2018 17:37:18 GMT
ETag: "2e77ad1dc6ab0b53a2996dfd4653c1c3"
Server: meinheld/0.6.1
Strict-Transport-Security: max-age=63072000
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
Vary: Accept-Encoding,Cookie
Age: 7

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>A basic webpage</title>
</head>
<body>
  <h1>Basic HTML webpage</h1>
  <p>Hello, world!</p>
</body>
</html>
```

Notificação de que o recurso foi movido permanentemente:

```
HTTP
HTTP/1.1 301 Moved Permanently
Server: Apache/2.4.37 (Red Hat)
Content-Type: text/html; charset=utf-8
Date: Thu, 06 Dec 2018 17:33:08 GMT
Location: https://developer.mozilla.org/
```

(Este é o novo link para o recurso; espera-se que o user-agent o acesse.)

Notificação de que o recurso solicitado não existe:



Códigos de status de resposta

Os **códigos de status HTTP** indicam se uma requisição HTTP específica foi completada com sucesso. As respostas são agrupadas em cinco classes:

- Respostas informativas (1xx)
- Respostas de sucesso (2xx)
- Redirecionamentos (3xx)
- Erros do cliente (4xx)
- Erros do servidor (5xx)

Alguns exemplos:

- **200**: OK. A requisição foi bem-sucedida.
- **301**: Movido Permanentemente. Significa que o URI do recurso solicitado foi alterado.
- **404**: Não Encontrado. O recurso solicitado não pôde ser localizado.